REMARKS

This amendment is responsive to the Office Action dated March 10, 2010. Applicant has amended claims 1, 24, 25 and 26. Claims 6 and 9-10 were previously cancelled. Claims 1-5, 7-8 and 11-26 are pending.

Claim Rejection Under 35 U.S.C. § 101

In the Office Action, the Examiner rejected claim 26 under 35 U.S.C. 101 based on an assertion that the claimed invention is directed to non-statutory subject matter. Applicant has amended claim 26 as suggested by the Examiner

Claim Rejection Under 35 U.S.C. § 112

In the Office Action, the Examiner rejected claims 1–5, 7, 8, and 11–26 under 35 U.S.C. 112, first paragraph, based on an assertion that the claims fail to comply with the written description requirement. Applicant has amended the claims as suggested by the Examiner to recite that the portion of the network message and the selected predefined string being compared have an equal number of characters. Applicant requests withdrawal of all rejections under 35 U.S.C. 112, first paragraph.

In the Office Action, the Examiner rejected claims 1–5, 7, 8, and 11–26 under 35 U.S.C. 112, second paragraph, based on an assertion that the claims omit essential steps. Although the Applicant respectfully disagrees, Applicant has nevertheless amended the claims as suggested by the Examiner.

Claim Rejection Under 35 U.S.C. § 103

The Examiner separately rejected the pending claims under six different combinations of prior art references. Applicant addresses each combination below and respectfully traverses the rejections to the extent such rejections may be considered applicable to the claims as amended. The applied references fail to disclose or suggest the inventions defined by Applicant's claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

Rejections based on first combination of prior art

In the Office Action, the Examiner rejected claims 1–3, 5, 8, 14–17, 19–20, and 22–26 under 35 U.S.C. 103(a) as being unpatentable over Smith et al. (US 2005/0246716) in view of Larson et al. (US 6,381,616). The Examiner also rejected claim 4 under 35 U.S.C. 103(a) as being unpatentable over Smith et al. in view of Larson et al. and in further view of Thinkage GCOS8 SS C Reference Manual, pages 1–71, 1996. The Examiner also rejected claims 7 and 11–13 under 35 U.S.C. 103(a) as being unpatentable over Smith et al. in view of Larson et al. and in further view of Abgrall et al. (US 2003/0037237). The Examiner rejected claim 18 under 35 U.S.C. 103(a) as being unpatentable over Smith et al. in view of Larson et al. and in further view of Kontio et al. (US 2005/0004875). The Examiner also rejected claim 21 under 35 U.S.C. 103(a) as being unpatentable over Smith in view of Larson and in further view of Slater et al. (US 6,654,796).

As a preliminary comment, the Examiner's rejection is very difficult to understand. In particular, the Examiner has cited certain paragraphs of Smith with little explanation and has cited the same paragraphs for each of the different steps of Applicant's claim 1.

Smith describes an application program interface (API) for application developers who build Web applications on a network platform. *Summary.* None of the paragraphs of Smith cited by the Examiner appear to describe any technique by which to actually carry out a comparison of two strings. In the rejection, the Examiner characterizes Smith as teaching that "http strings are case insensitive and http strings [sic] comparison contain operations." *OA, pg. 13.* The Examiner also relies on Smith as teaching that "http strings comparison [sic] produce result [sic]". *OA, pg. 12.* Notwithstanding these irrelevant assertions, Smith in view of the other references do not teach or suggest the method set forth in claim 1. In particular, paragraphs [23268]-[23721] appear to list various functions of the API that can be invoked so as to compare strings or perform other string related operations, however, Smith does not appear to describe how these operations are carried out.

The Examiner admits that Smith does not teach any mechanism for comparing two strings in stating that "Smith-Microsoft does not limit its string comparison to certain techniques." *OA, pg. 13.*

The Examiner then looks to Larson at cols. 10-14 to address the deficiencies of Smith. Larson describes a system and method for speeding up heterogeneous data access using predicate

conversion. *Larson, Title.* In particular, Larson describes a system and method for modifying or converting predicates in a query so that the predicates can be evaluated by an external or remote source that uses a different collating sequence, thereby effectively reducing the amount of data sent over a network. *Larson, Abstract.* In other words, Larson describes a mechanism for modifying the form of a data query. *See, e.g., Larson, claim 1.* Larson provides a database system that uses a conversion table generator using a conversion rule to convert the query. *Larson, Summary.* The portions of Larson relied upon by the Examiner explain in detail the operations applied by the conversion rule to modify the query. The cited portions do not teach any technique for actually comparing strings.

In sum, with respect to claim 1, the Examiner admits that Smith does not describe any techniques for comparing strings and instead relies on Larson which describes applying operations to modify the form of a database query. The cited references to do not, either individually or in combination, teach or suggest any mechanism for carrying out a comparison of two strings, let alone in accordance with the elements set forth in Applicant's claims, as amended.

As such, Smith in view of Larson fails to teach or suggest many elements of claim 1, as amended. For example, Smith in view of Larson fails to teach or suggest dividing the unknown string and the selected predefined string into segments, wherein the segments of the unknown string and the segments of the selected predefined string have an equal number of the characters. Smith in view of Larson also fails to teach or suggest performing a bitwise exclusive OR operation between the ASCII binary representation of each of the segments of the unknown string and the ASCII binary representation of each of the segments of the selected predefined string, performing a bitwise operation between a predefined flag and each result of the exclusive OR operations, wherein the predefined flag has an equal number of characters as the result of the exclusive OR operation. Moreover, Smith in view of Larson fails to teach or suggest comparing the predefined flag and each result of the bitwise operations to produce an indication for a case-insensitive string match, wherein the indication for the case-insensitive string match indicates whether all the characters of the segments of the unknown string within the network message match all the corresponding characters of the segments of the selected predefined string so as to match one of the known headers of the network communication protocol.

Applicant points out that the Examiner's general assertions that some network systems process network communications and manipulate strings do not provide substantial evidence that the particular techniques for performing the string comparisons in the manner described and claimed by the Applicant are unpatentable over such systems. The Examiner appears to be unwilling to consider the actual steps set forth in any of Applicant's claims with respect techniques for performing string comparisons in view of the teachings of the prior art. This is clearly erroneous. Indeed, the Examiner appears to rely on impermissible hindsight to combine these references in a manner that reaches the presently claimed invention. The U.S. Supreme Court cautions in *KSR* that "A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon ex post reasoning."[1] Moreover, "[i]t is impermissible to use the claimed invention as an instruction manual or 'template' to piece together the teachings of the prior art so that the claimed invention is rendered obvious."[2] That the applied references provide no teachings to suggest how the above noted conversions can be accomplished is indicative of the use of impermissible hindsight to the extent that it appears the Examiner has relied on the claimed invention as a "template" by which to piece together teachings from the disparate technologies disclosed by the applied references to render the presently claimed invention obvious. This is impermissible.

Rejections based on second combination of prior art

The Examiner also rejected claims 1–3, 5, 8, 14–17, 19–20, and 22–26 under 35 U.S.C. 103(a) as being unpatentable over Knouse et al. (US 7,249,369) in view of James et al. (US 6,523,108). The Examiner also rejected claim 4 under 35 U.S.C. 103(a) as being unpatentable over Knouse in view of James and in further view of Thinkage GCOS8 SS C Reference Manual, pages 1–71, 1996. The Examiner also rejected claims 7 and 11–13 under 35 U.S.C. 103(a) as being unpatentable over Knouse in view of James and in further view of Abgrall. The Examiner also rejected claim 18 under 35 U.S.C. 103(a) as being unpatentable over Knouse in view of James and in further view of Kontio. The Examiner also rejected claim 21 under 35 U.S.C. 103(a) as being unpatentable over Knouse in view of James et al. and in further view of Slater.

---

[1] *KSR Int'l Co. v. Telejlex, Inc.*, 550 U.S. 398,421 (2007).
[2] *In re Fritch*, 972 F.2d 1260, 1266 (Fed. Cir.1992) (*citing In re Gorman*, 933 F.2d 982,987 (Fed. Cir. 1991)).

Knouse describes a system capable of matching sets of authentication, authorization, and auditing rules contained in policies to resources in an access system using POST data received in HTTP POST requests. *Knouse, Summary*. Knouse suffers from the same deficiencies of Smith as discussed above. In particular, the Examiner relies on Knouse merely as describing a system that operates on HTTP headers. The Examiner admits that Knouse describe not specific technique for performing string comparisons with the HTTP headers. *Office Action, pg. 25, stating "Kouse-Oracle does not limit its string comparison to certain techniques.")* The Examiner looks to James at cols. 3, 4 to provide a "bitwise technique for string comparison" so as to address the deficiencies of Smith. *Id.*

However, James does not describe a bitwise technique for comparing strings. Instead, James describes a method for extracting a string of bits from a binary bit string. *James, Title*. According to James, the method for extracting the bits includes deposit and extract instructions that have an opcode, a source address, a destination address, a shift number, and a K-bit mask string. *James, Abstract*. The portions of James relied upon by the Examiner (cols. 3, 4) describe an instruction set applied by a CPU including a shift instruction, a rotation instruction, a mask instruction. In col. 4, James describes an instruction for extracting a portion of a string and placing that extracted portion into a destination string. Contrary to the Examiner, James does not describe any mechanism for comparing two different strings.

In sum, with respect to claim 1, the Examiner admits that Knouse does not describe any technique for comparing strings and relies on James which describes applying operations to extract a portion of a string. The cited references to do not, either individually or in combination, teach or suggest any mechanism for carrying out a comparison of two strings, let alone in accordance with the elements set forth in Applicant's claims.

As such, Knouse in view of James fails to teach or suggest many elements of amended claim 1. For example, Knouse in view of James fails to teach or suggest dividing the unknown string and the selected predefined string into segments, wherein the segments of the unknown string and the segments of the selected predefined string have an equal number of the characters. Knouse in view of James also fails to teach or suggest performing a bitwise exclusive OR operation between the ASCII binary representation of each of the segments of the unknown string and the ASCII binary representation of each of the segments of the selected predefined string, performing a bitwise operation between a predefined flag and each result of the exclusive

OR operations, wherein the predefined flag has an equal number of characters as the result of the exclusive OR operation. Moreover, Knouse in view of James fails to teach or suggest comparing the predefined flag and each result of the bitwise operations to produce an indication for a case-insensitive string match, wherein the indication for the case-insensitive string match indicates whether all the characters of the segments of the unknown string within the network message match all the corresponding characters of the segments of the selected predefined string so as to match one of the known headers of the network communication protocol.

Rejections based on the third combination of prior art

The Examiner also rejected claims 1–3, 5, 8, 14–17, 19–20, and 22–26 under 35 U.S.C. 103(a) as being unpatentable over Reiche et al. (US 6,092,196) in view of Geist (US 5,329,598).

Reiche describes authentication server for use in a data network. The portions of Reiche cited by the Examiner (cols. 5-6) describe a user using an "HTTP data exchange session" with a web browser to access a resource. [3] At this time the user is redirected to a central authentication server that determines whether the user is authorized to access the resource. Reiche states that the authentication server receives a response and compares data of the response to entries in a database of user IDs and passwords.[4] Reiche does not explain at all how this comparison is carried out. The Examiner admits this deficiency by stating that "Reiche-Nortel-Networks does not limit its string comparison to certain technique [sic]." *OA, pg. 35.*

The Examiner looks to Geist to address the deficiencies of Smith. The Examiner characterizes Geist as disclosing "well- known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., Figures 9A-9D and its description." However, Geist does not describe techniques for comparing case-insensitive strings as asserted by the Examiner. Geist describes a method for automatically identifying and correcting errors in electronically stored character strings input from handwritten character strings. *Geist, Abstract.* The input character strings are compared to a predetermined list of correct character strings by dividing the input character string and each of the correct character strings into at least one character string fragment. Each character string fragment or set of

---

[3] Reiche, Summary.
[4] *Id.*

13

character string fragments is formed by applying at least one different fragmentation submethod to the character string. The corresponding fragments from the input character string and the correct character strings are then compared in turn to determine if there is an exact match. That is, Geist does not teach a technique for applying a case-insensitive match but instead determines wether the corresponding fragments exactly match.

FIG. 2B of Geist shows the compare subroutine. As shown, pointers are used to reference respective characters and an XOR operation is applied to determine if there is an exact match between the characters. If not, a "missed" error value is determined. Thus, Geist does not teach any mechanism for a case-insensitive match and, in fact, similar letters of a different case (e.g., "A" and "a") would be treated as miss and not a match. FIG. 2B is reproduced below.

FIG.2B
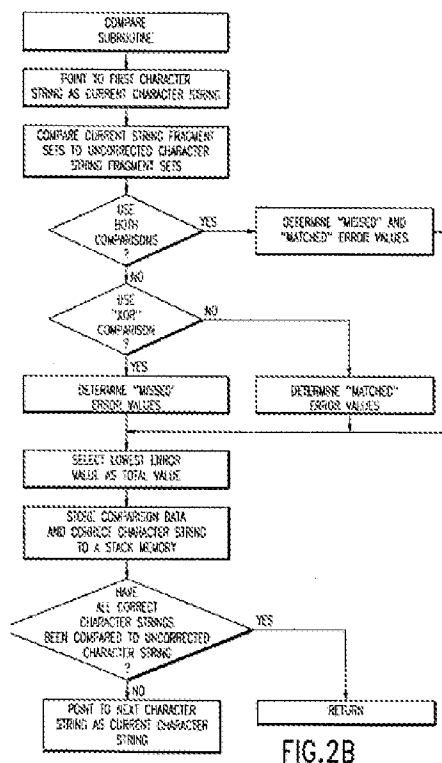
As such, Reiche in view of Geist fails to teach or suggest many elements of amended claim 1. For example, Reiche in view of Geist fails to teach or suggest performing a bitwise operation between a predefined flag and each result of the exclusive OR operations, wherein the predefined flag has an equal number of characters as the result of the exclusive OR operation. Quit the contrary, Reiche does not describe any technique for comparing strings and Geist

14

describes an exact match comparison that applies only and exclusive OR operation between the characters themselves. Neither suggest the use of an additional predefined flag. Moreover, Knouse in view of James fails to teach or suggest comparing the predefined flag and each result of the bitwise operations to produce an indication for a case-insensitive string match, wherein the indication for the case-insensitive string match indicates whether all the characters of the segments of the unknown string within the network message match all the corresponding characters of the segments of the selected predefined string so as to match one of the known headers of the network communication protocol.

Rejections based on forth combination of prior art

The Examiner also rejected claims 1–3, 5, 8, 14–17, 19–20, and 22–26 under 35 U.S.C. 103(a) as being unpatentable over Aviani, Jr. et al. (US 6,532,493) in view of Khieu (US 5,381,127). The Examiner also rejected claim 4 under 35 U.S.C. 103(a) as being unpatentable over Aviani, Jr. et al. (US 6,532,493) in view of Khieu (US 5,381,127) and in further view of Thinkage GCOS8 SS C Reference Manual, pages 1–71, 1996. The Examiner also rejected claims 7 and 11–13 under 35 U.S.C. 103(a) as being unpatentable over Aviani, Jr. et al. (US 6,532,493) in view of Khieu (US 5,381,127) and in further view of Abgrall et al. (US 2003/0037237). The Examiner also rejected claim 18 under 35 U.S.C. 103(a) as being unpatentable over Aviani, Jr. et al. (US 6,532,493) in view of Khieu (US 5,381,127) and in further view of Kontio et al. (US 2005/0004875). The Examiner also rejected claim 21 under 35 U.S.C. 103(a) as being unpatentable over Aviani, Jr. et al. (US 6,532,493) in view of Khieu (US 5,381,127) and in further view of Slater etal. (US 6,654,796).

Aviani describes a network caching system.[5] When requested content is not in the cache, the cache attempts to establish a connection to the destination server to retrieve the content. At this time, the cache adds certain information to the HTTP header of the request.[6] The cited portions of Aviani (col. 6, ln. 20 – co. 8, ln. 26), describe the network cache modifying the HTTP header so that the original client platform is identified for any subsequent caching systems encountered by the request. In relevant part, Aviani states that, upon reception of a new request

---

[5] Aviani, Summary.
[6] *Id.*

15

by the same caching system, a client/server pair identified by the HTTP header is compared to a resident bypass list.[7] Aviani does not explain at all how this comparison is carried out.

The Examiner admits that Aviani does not teach any mechanism for comparing two strings in stating that "Aviani-Cisco-Technology does not limit its string comparison to certain technique [sic]." *OA, pg. 41.* The Examiner then looks to Khieu to address the deficiencies of Smith. The Examiner characterizes Khieu as disclosing "well- known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 4 and its description." However, Khieu does not describe techniques for comparing case-insensitive strings as asserted by the Examiner.

Khieu describes a high speed, static BiCMOS cross-unit comparator circuit is described which incorporates the use of a single stage hit-miss detection circuit capable of executing within a single gate delay. FIG. 4 is a block diagram showing a signal generated at input 90 which triggers two relatively independent circuits, 91 and 92, to process information. The comparator circuit 102 is then used to determine whether the data 93 produced by circuit 91 is the same as the data 94 produced by circuit 92. The comparator circuit 102 comprises two stages. The first stage is a combined XOR/XNOR gate 97.
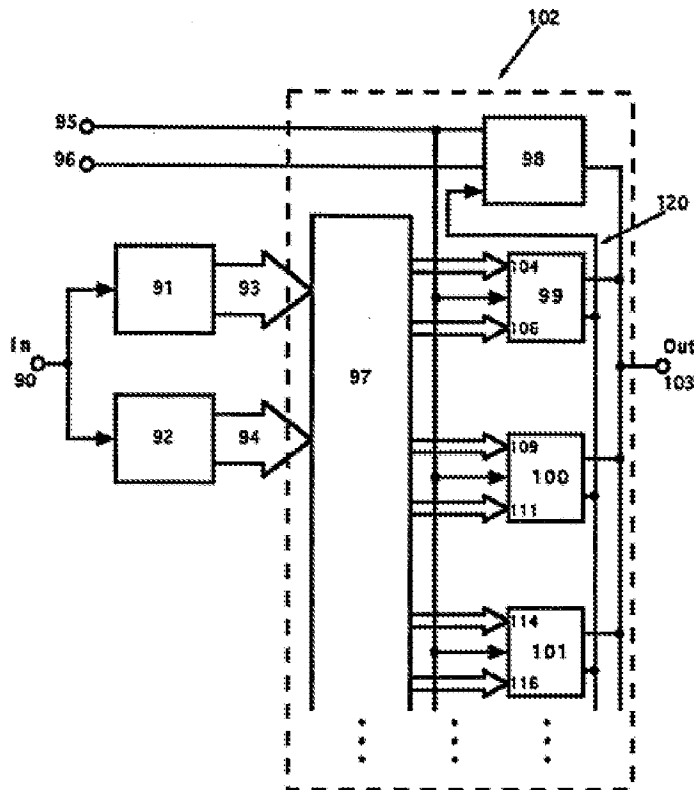
---

[7] Aviani, col. 7, ll. 33-35.

Figure 4

Thus, in sum, Khieu describes a circuit for performing an exact match between data 93 and data 94. In sum, with respect to amended claim 1, the Examiner admits that Aviani does not describe any techniques for comparing strings and instead relies on Khieu which describes a circuit for determining an exact match between data. The cited references to do not, either individually or in combination, teach or suggest any mechanism for carrying out a case-insensitive comparison of two strings, let alone in accordance with the elements set forth in Applicant's claims.

As such, Aviani in view of Khieu fails to teach or suggest many elements of amended claim 1. For example, Aviani in view of Khieu fails to teach or suggest dividing the unknown string and the selected predefined string into segments, wherein the segments of the unknown string and the segments of the selected predefined string have an equal number of the characters. Aviani in view of Khieu also fails to teach or suggest performing a bitwise exclusive OR operation between the ASCII binary representation of each of the segments of the unknown

17

string and the ASCII binary representation of each of the segments of the selected predefined string, performing a bitwise operation between a predefined flag and each result of the exclusive OR operations, wherein the predefined flag has an equal number of characters as the result of the exclusive OR operation. Moreover, Aviani in view of Khieu fails to teach or suggest comparing the predefined flag and each result of the bitwise operations to produce an indication for a case-insensitive string match, wherein the indication for the case-insensitive string match indicates whether all the characters of the segments of the unknown string within the network message match all the corresponding characters of the segments of the selected predefined string so as to match one of the known headers of the network communication protocol.

Rejections based on fifth combination of prior art

The Examiner also rejected claims 1 and 24–26 under 35 U.S.C. 103(a) as being unpatentable over Wein et al. (US 7,240,100) in view of Chopra et al. (US 6,631,466).

Wein describes a content delivery network that serves content to customers. The cited portions of Wein (col. 5, ln. 10 – col. 6, ln. 59) state that the content delivery uses a header in an HTTP request to determine the appropriate metadata to locate and server.[8] The cited portions of Wein do not describe how the content delivery network processes the header and Applicant can find no relevant teachings with respect to Applicant's claims. The Examiner admits this deficiency by stating that "Wein-Nortel-Networks does not limit its string comparison to certain technique [sic]." *OA, pg. 50.*

The Examiner looks to Chopra to address the deficiencies of Wein. The Examiner characterizes Chopra as disclosing "well- known techniques of comparing case-insensitive strings without converting the strings to a common case, e.g., usage of XOR, etc., figures 7A and 7B and its description." However, Chopra does not describe techniques for comparing case-insensitive strings as asserted by the Examiner. Chopra describes a method for searching a data packet for one or more patterns such as a string or a series of strings. *Chopra, Summary.*

Like Khieu discussed above, Chopra describes a circuit for performing an exact match. See Figures 7A, 7B. In sum, with respect to claim 1, the Examiner admits that Wein does not describe any techniques for comparing strings and instead relies on Chopra which describes a

---

[8] Wein, col. 6, ll. 54-60.

circuit for determining an exact match between data. The cited references to do not, either individually or in combination, teach or suggest any mechanism for carrying out a case-insensitive comparison of two strings, let alone in accordance with the elements set forth in Applicant's claims.

As such, Wein in view of Chopra fails to teach or suggest many elements of amended claim 1. For example, Wein in view of Chopra fails to teach or suggest dividing the unknown string and the selected predefined string into segments, wherein the segments of the unknown string and the segments of the selected predefined string have an equal number of the characters. Wein in view of Chopra also fails to teach or suggest performing a bitwise exclusive OR operation between the ASCII binary representation of each of the segments of the unknown string and the ASCII binary representation of each of the segments of the selected predefined string, performing a bitwise operation between a predefined flag and each result of the exclusive OR operations, wherein the predefined flag has an equal number of characters as the result of the exclusive OR operation. Moreover, Wein in view of Chopra fails to teach or suggest comparing the predefined flag and each result of the bitwise operations to produce an indication for a case-insensitive string match, wherein the indication for the case-insensitive string match indicates whether all the characters of the segments of the unknown string within the network message match all the corresponding characters of the segments of the selected predefined string so as to match one of the known headers of the network communication protocol.

Rejections based on the sixth combination of prior art

The Examiner also rejected claims 1 and 24–26 under 35 U.S.C. 103(a) as being unpatentable over Ciles et al. (US 6,986,047) in view of James.

Giles describes methods for enabling access to a semi-trusted web server.[9] The cited portion of Giles (col. 5, ln. 12 – col. 6, ln. 58) describes correlating a client IP address and HTTP header information to a client credential. The cited portions of Giles do not describe how the correlation is performed and Applicant can find no relevant teachings with respect to Applicant's claims. As explained above, James does not describe a bitwise technique for comparing strings.

_____

[9] Giles, Abstarct.

19

Instead, James describes a method for extracting a string of bits from a binary bit string. *James, Title*. According to James, the method for extracting the bits includes deposit and extract instructions that have an opcode, a source address, a destination address, a shift number, and a K-bit mask string. *James, Abstract*. The portions of James relied upon by the Examiner (cols. 3, 4) describe an instruction set applied by a CPU including a shift instruction, a rotation instruction, a mask instruction. In col. 4, James describes an instruction for extracting a portion of a string and placing that extracted portion into a destination string. Contrary to the Examiner, James does not describe any mechanism for comparing two different strings.

For at least these reasons, the Examiner has failed to establish a prima facie case for non-patentability of Applicant's claims under 35 U.S.C. 103(a). Withdrawal of this rejection is requested.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date: _____July 12, 2010_____   By: ____/Kent J. Sieffert/____
SHUMAKER & SIEFFERT, P.A.   Name: Kent J. Sieffert, Reg. No.: 41,312
1625 Radio Drive, Suite 300
Woodbury, Minnesota 55125
Telephone: 651.286.8341
Facsimile: 651.735.1102